# Agile Software Development Education in Hybrid Learning Environments

Nirnaya Tripathi*, Woubshet Behutiye and Minna Isomursu

M3S research unit, University of Oulu, Oulu, Finland

**Abstract**

Educational institutions have begun implementing a hybrid teaching approach that incorporates both online and in-person learning to provide students with greater flexibility in learning. Although a significant amount of literature is available to help understand how software engineering education can be conducted in hybrid settings, empirical research on how Agile software development (ASD) can be effectively taught in such settings is lacking. This study aims to address such a gap by presenting a case of software processes and the Agile method course in hybrid settings, providing knowledge to help educators align with the current needs of the software industry and educational institutions. The authors report their three years of experience conducting a hybrid course, and they share students perceptions of hybrid teaching. The study discusses a course design that uses constructive alignment to achieve the intended learning outcomes in a hybrid teaching setting. It also explores the challenges faced while teaching in such settings and provides corresponding recommendations. The research includes a data analysis revealing students satisfaction with the overall course. By providing a comprehensive analysis and practical recommendations, this study aims to advance ASD education in hybrid settings, aligning academic efforts with evolving trends in the field of software.

**Keywords**

hybrid learning, education, agile methods, empirical research, software process

## 1. Introduction

In the software industry today, software development teams working in remote settings have become common. Similarly, academic institutions have shifted to hybrid classes, which allow in-person and online learning for students. These developments have led universities to focus on software engineering education to teach students in hybrid settings. [1], [2] Additionally, in the ASD course, students must be familiar with the application of Agile practices and the use of tools that support working remotely and collaboratively.

**Research Problem.** Recent studies [3],[2],[1] suggest that hybrid teaching in software engineering courses can provide various opportunities. While there are online studies available on Agile software education [4],[5], limited research on hybrid contexts has been conducted. Therefore, further empirical research is needed to explore how hybrid settings affect course design and student learning. The following aspects should be considered in such studies:

- Teaching methods that integrate both online and in-person approaches
- More accessible Agile education for international and remote students
- Integration of Agile project management tools into the coursework
- Adopting the industry trend of hybrid software development

**Research Approach.** This study aims to enhance the understanding of hybrid teaching methodologies for Agile methods and software process models using the constructive alignment approach and empirical data collected from students. This task involves recognizing the distinct characteristics, issues, and prospects of hybrid teaching. It focuses on exploring the following research questions to gain insights into the topic:

- Q1. How do students assess the effectiveness of learning outcomes and teaching methods in hybrid courses?
- Q2. What are some strategies to minimize issues that may arise in the teaching methods adopted in hybrid course environments?

This study provides significant insights into hybrid teaching by answering the above questions based on the authors experiences in teaching the course (Section 3) and by systematically analyzing

TKTP 2024: Annual Doctoral Symposium of Computer Science, 10.-11.6.2024 Vaasa, Finland

*Corresponding author.

Nirnaya.Tripathi@oulu.fi (N. Tripathi); Woubshet.Behutiye@oulu.fi (W. Behutiye); Minna.Isomursu@oulu.fi (M. Isomursu)

data collected from students feedback over three years of teaching the course (Section 4). The findings can potentially assist educators in improving their teaching methods and enhancing students overall learning experiences.

**Study Contributions.** This study contributes to Agile software development (ASD) education in a hybrid setting in the following ways:

- Using pedagogical techniques, such as constructive alignment, in course design (Section 3)
- Applying teaching methods to instruct students both face to face and online (Section 3.2)
- Simulating ASD in physical and remote settings for students project work
- Presenting students experiences with the learning outcomes (LOs), course content, teaching methods, and learning assessment (Section 4)
- Highlighting issues in teaching methods in hybrid learning environments with corresponding recommendations (Section 5.2)

This paper is organized into the following main sections. In Section 2, we present the background literature. In Section 3, we share our Agile software course in a hybrid setting. In Section 4, we provide the empirical evaluation of the course from students perspectives. Section 5 discusses the interrelationships between teaching methods and LOs, the influence of a hybrid setting on teaching methods, and the studys validity. Lastly, Section 6 concludes the study.

## 2. Background Literature

Agile software development methodologies are widely used in software engineering, and educators and researchers are interested in implementing them in the educational context to teach students. Teaching Agile methodologies in the classroom while using them through project work could foster students learning. Previous literature [6] has discussed the application of Agile values to educational settings and the integration of Agile methodologies into education, emphasizing iterative learning, continuous feedback, and collaboration. Agile tools, such as user stories and burndown charts, have been proposed for pedagogical use [7]. The effectiveness of Agile strategies in online higher education in terms of team regulation and project management was examined by Noguera et al. in [4]. The authors in [8] discussed the challenges that students encounter when implementing Agile practices in a course and offered recommendations aiming to enhance Agile practices in universityindustry projects.

Constructive alignment is a teaching approach that involves aligning teaching methods and assessment tasks with the intended learning objectives, as discussed by Biggs in [9]. This approach has been successfully applied in various disciplines. Hypponen et al. in [10] provided an overview of the types of teaching methods and assessment tasks that can be used. Constructive alignment has also been utilized in teaching software engineering, specifically in the areas of software architecture and testing in the literature [11], [12]. Cain and Babar in [12] presented two case studies demonstrating constructive alignment in software engineering. These studies highlighted the importance of feedback and assessment in the course. Hynninen et al. [11] discussed the process of constructively aligning software testing education. The authors proposed an initial design for a software testing course based on the results of an industry survey.

Hybrid learning combines traditional classroom instruction with online learning activities to provide flexible and personalized learning experiences. Research has been conducted on software engineering education in this setting. For example, the authors in [2] provided valuable guidelines for educators based on experiences and lessons learned from hybrid teaching. Another article [1] explored the use of project-based learning (PBL) in a hybrid course that included both online and in-person students. The study discussed how PBL, in which the student takes the center stage in the learning process while the instructor acts as a facilitator, helps students develop both soft and hard skills in software development. The authors in [3] and [5] proposed a study on the use of communication software and Agile project management methodologies to prepare students for remote software development positions after graduation. These articles offered methods for enhancing higher education in both the classroom and remote settings.

## 3. Course Description

The University of Oulu offers bachelors and masters degree programs in software engineering and information systems. Additionally, it offers degree programs such as Software and Systems Development in the Global Environment. Graduating students are required to take a course on professional

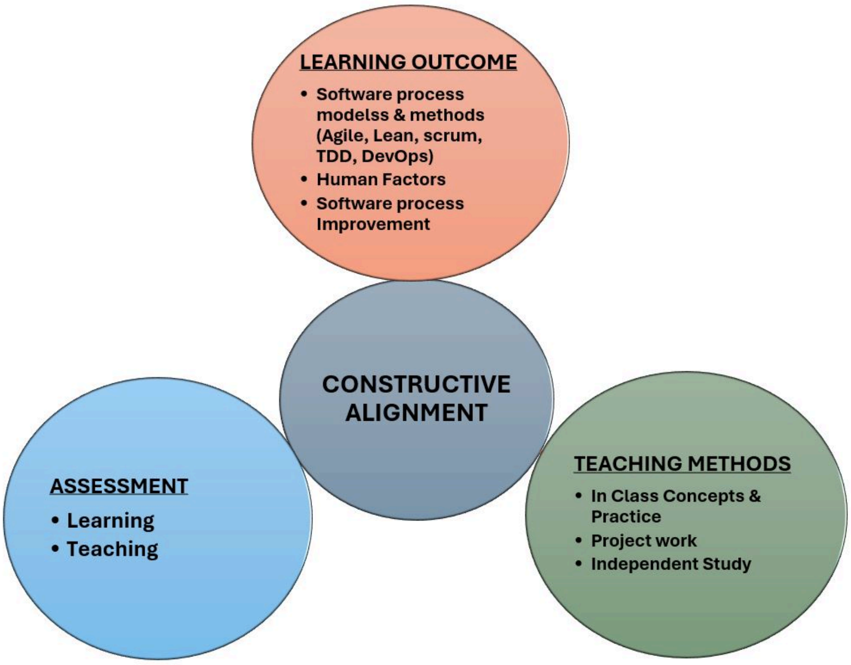

LEARNING OUTCOME
Software process modelss & methods (Agile, Lean, scrum, TDD, DevOps)
Human Factors
Software process Improvement
CONSTRUCTIVE ALIGNMENT
ASSESSMENT
Learning
Teaching
TEACHING METHODS
In Class Concepts & Practice
Project work
Independent Study

**Figure 1:** Course based on Constructive alignment method

software engineering processes and human factors [1], which can be attended in person.

**Motivation for hybrid teaching.** The course is also offered to students through other channels, such as the Open University [2] and the FITech Network University [3]. Through these avenues, students are allowed to participate entirely online. Effective teaching requires the use of appropriate working methods to achieve quality LOs [10]. As the course is offered through different channels, several students prefer to attend classes in person, but some students work full-time in industries or live in other cities or countries and prefer to participate remotely. As a result, the hybrid learning mode was selected as a working method because it combines online educational materials and opportunities for interaction online with traditional place-based classroom methods.

**Course setting.** The course was designed on the basis of the constructive alignment method (see Figure 1), with a focus on aligning teaching methods (section 3.2) and assessments (Section 3.3) to the intended LOs (Section 3.1). The course has been running for three years. In school years 2021-2023, a total of 67-72 students enrolled. The course has 5 ECTS credits, with a workload of 135 hours over two months. It consists of 66 hours of in-class/independent work and 69 hours of project work. Grading is 50% each for independent work and group projects. Figure 2 shows the course timeline and content.

## 3.1. Learning Outcomes

Learning outcomes refer to the knowledge, skills, abilities, or values that a student is expected to acquire by the end of a course. The course has several objectives. First, it aims to familiarize students with the various software development process models and Agile methods used in the software industry (LO1, LO2). Second, it seeks to teach students about human factors, as software development is a human-driven process (LO3). Third, the course enables students to be aware of their strengths and weaknesses as software development engineers

[1] Course https://opas.peppi.oulu.fi/en/course/811373A/10780?period=2023-2024
[2] Open University https://joy.oulu.fi/en/education-search/professional-software-engineering-processes-and-human-factors-open-uni-0
[3] FITech https://fitech.io/en/studies/professional-software-engineering-processes-and-human-factors/

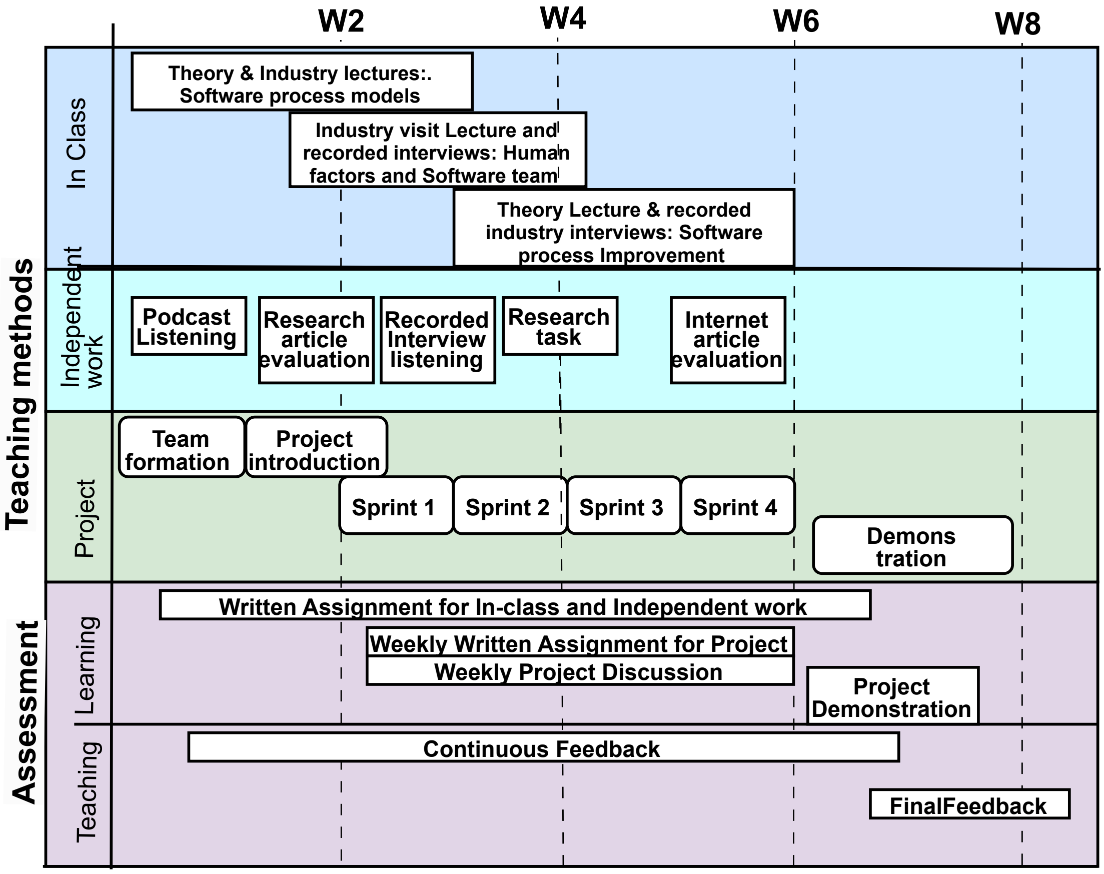

W2
W4
W6
W8
Teaching methods
In Class
Theory & Industry lectures:. Software process models
Industry visit Lecture and recorded interviews: Human factors and Software team
Theory Lecture & recorded industry interviews: Software process Improvement
Independent work
Podcast Listening
Research article evaluation
Recorded Interview listening
Research task
Internet article evaluation
Project
Team formation
Project introduction
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Demons tration
Assessment
Learning
Written Assignment for In-class and Independent work
Weekly Written Assignment for Project
Weekly Project Discussion
Project Demonstration
Teaching
Continuous Feedback
FinalFeedback

**Figure 2:** Course Timeline

and managers by considering human factors (LO4). Fourth, the course aims to teach students about software process improvement (SPI) techniques, as issues and bottlenecks in the process can affect software quality and project success (LO5). The LOs are clearly outlined to achieve these objectives. The relevant course content and topics are provided in Table 1.

## 3.2. Teaching Methods

Effective teaching methods are crucial in helping students learn and stay motivated. The suitability of a method, its application, student engagement, and teaching resources all influence the effectiveness of a teaching method. In our hybrid learning mode, we included in-class concepts and practice, independent study work, and project work.

### 3.2.1. In-Class Concepts and Practice

In-class concepts and practice is an educational approach that combines theoretical concepts with practical applications in the classroom. Class and exercise rooms are reserved for in-person students, while communication platforms, such as Zoom, can be used by remote students to maintain flexibility and accessibility for hybrid learning. Despite the challenges of balancing attention between in-person and remote students, we ensured that the physical classroom was equipped with adequate audio and video capabilities to support remote participants. We used a dedicated microphone and camera to capture lecture presentations and arranged them in such a way that instructors could maintain eye contact with both in-person and remote students. We utilized Moodle as a centralized platform for scheduling and notifications, as well as for storing learning materials, such as videos and slides. By pro-

**Table 1**
Overview of the learning outcomes and topics covered

| Learning Outcome | Course Content |
|---|---|
| LO1. To recognize and describe software development process models<br>LO2. To evaluate and compare software development process models and their applicability in different contexts | - Software engineering evolution, and Software Engineering Body of Knowledge and its knowledge areas<br>- Traditional software development models, such as the waterfall and V-model and the linear and sequential approaches<br>- Agile and lean software development, Scrum, Kanban, test-driven development, continuous deployment, and DevOps |
| LO3. To take human factors into account when planning and operating during professional software development<br>LO4. To analyze their own strengths and improvement areas as software engineers in order to see opportunities for development | - Team dynamics, diversity, and cultural considerations within software teams<br>- Human factors in software development at the individual, team, and organizational levels<br>- Hiring process emphasis on degrees and on practical experiences and technical skills |
| LO5. To participate in systematic efforts toward process improvement in software development organizations | - Software process improvement history and plan-do-check-act<br>- Software process assessment and standards: Capability Maturity Model Integration (CMMI), ISO 15504, and Automotive SPICE<br>- Quality improvement paradigm with IEC 33001, Six Sigma/Kaizen |

viding learning resources before and after lectures, we ensured equal access for all students. This accessibility fostered a conducive learning environment, allowing students to navigate the course content at their own pace.

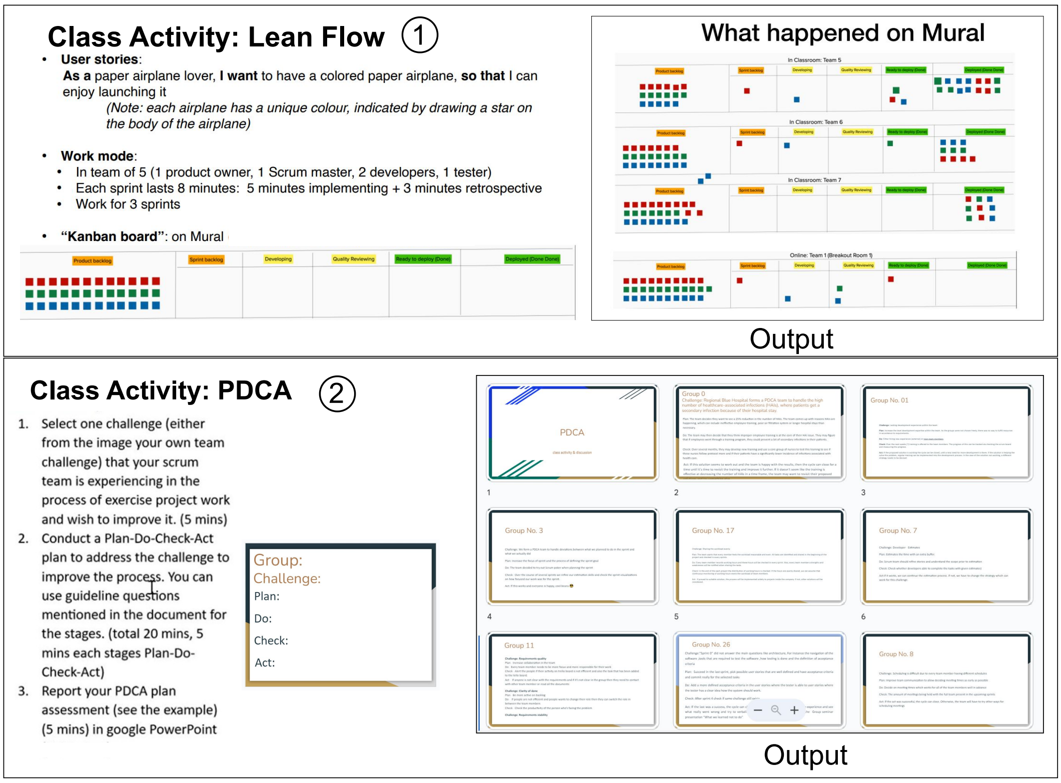
**Lecture.** This is a popular teaching method in which teachers present information to students in an organized manner to help create connections between different topics. The theory lecture topics were aligned with the LOs and course contents, as mentioned in Table 1. Experts in Agile and lean software development from the software industry, including project managers, DevOps specialists, and product owners, discussed various ways to customize Agile and lean processes and make them fit organizational needs. Experts were involved to share their ideas on adapting to the evolving needs of the software industry. Hybrid work was also discussed, emphasizing the benefits of having a diverse team with varied skills and cultural perspectives. During the session, technical issues with platforms, such as Zoom, can sometimes disrupt hybrid learning, so links to solutions were shared with the students. Some video interviews were also conducted with experts to highlight the importance of software process methods and enable faster product delivery, reduced complexity, and a culture of continuous learning. These were recorded and shared with the students as independent study material. Practical cases were mentioned by experts to illustrate these topics.

- Role of software in emerging digital services
- Characteristics of a professional software engineer
- Interdependence of software engineering managers and teams
- Importance of social skills and team activities in hiring decisions

**Class activity and quizzes.** Passive listening during hybrid session lectures can lead to decreased attention. Thus, we implemented varying lecturing styles, such as conducting class activities during a lecture and using interactive tools, such as polls, quizzes, and breakout rooms, to ensure equal engagement for both in-person and remote students. These activities can encourage students to participate. Some examples of class activities done in the hybrid setting include solving the Scrum framework puzzle, applying the plan-do-check-act (PDCA) plan to a process problem using PowerPoint online, and creating a small artifact to demonstrate the concept of lean flow with Scrum practices using the Mural tool as backlog (see Figure 3).

### 3.2.2. Independent Work

Independent work is a teaching method that allows students to complete assignments without direct supervision. In this approach, students individually engage in various learning activities divided into weekly milestones throughout six weeks, such as analyzing concepts in research articles, conducting small research and exercise tasks, listening to recorded lectures and interviews, and reflecting on their learning (see Figure 2b). Instructors provide



Class Activity: Lean Flow ①

- **User stories**:
  **As a** paper airplane lover, **I want** to have a colored paper airplane, **so that** I can enjoy launching it
  *(Note: each airplane has a unique colour, indicated by drawing a star on the body of the airplane)*
- **Work mode**:
  - In team of 5 (1 product owner, 1 Scrum master, 2 developers, 1 tester)
  - Each sprint lasts 8 minutes: 5 minutes implementing + 3 minutes retrospective
  - Work for 3 sprints
- **"Kanban board"**: on Mural

What happened on Mural

Output

Class Activity: PDCA ②

1. Select one challenge (either from the image your own team challenge) that your scrum team is experiencing in the process of exercise project work and wish to improve it. (5 mins)
2. Conduct a Plan-Do-Check-Act plan to address the challenge to improve the process. You can use guideline questions mentioned in the document for the stages. (total 20 mins, 5 mins each stages Plan-Do-Check-Act)
3. Report your PDCA plan assessment (see the example) (5 mins) in google PowerPoint

Group:
Challenge:
Plan:
Do:
Check:
Act:

Output

**Figure 3:** Class activity combining in-person and online interaction

clear instructions and guidelines for conducting the work. However, in this study, some challenges were faced, such as remote students struggling with digital literacy or motivation to complete the tasks because of isolation. To address this, the instructors reserved dedicated time slots for online meetings. Below were some tasks given:

- The students assessed software projects suitable for waterfall, iterative, or Agile development cycles using article [13], lecture material, and a podcast.
- After watching the industry lecture on Agile, Scrum, test-driven development, and DevOps, the students reflected on the knowledge they acquired and implemented it to improve their project work.
- To improve the job application process, the students analyzed an article on the technical interview process [14].
- The students read and reflected on the paper [15] regarding human factors and their influences on software development.
- The students reflected on software quality improvement approaches by watching a video and reading an article on Six Sigma, Kaizen, and CMMI.

### 3.2.3. Project Work

Project work involves linking theoretical knowledge and practical actions. Participants work actively on a project and retrieve information related to the projects objectives. In the course, the instructors formed multicultural teams of students who worked together in a hybrid setting. The learning objectives for the project work were as follows:

1. Implement ASD practices using supported tools
2. Create a backlog of requirements and develop a prototype
3. Work together (physical and remote modes) in teams of five members with roles such as the product owner, Scrum master, UX designer, and developer

During the projects beginning, the student groups were divided into Scrum teams. Instructors acted as customers during four-week sprints, in which the students had to provide project plans, prototypes, and demonstrations as deliverables. This helped apply Scrum theoretical knowledge to the practical prototype development project. The project involved both in-person and remote students who used digital tools, such as Jira, Miro, and Trello, to collaborate. The teams used ceremonies, such as backlog refinement, sprint planning, daily standups, sprint reviews, and sprint retrospectives. By participating in sprint reviews, they reflected on work processes, identified areas for improvement, and implemented changes in subsequent sprints. During the sprint reviews, the instructor implemented icebreaking activities to build trust and ensure that everyone felt valued. Remote students used tools, such as Doodle, to find common meeting times with on-campus students. Jira and Trello were used to delegate tasks, set deadlines, and track progress. These tools ensured that all members were aware of their responsibilities and the projects overall progress.

## 3.3. Learning and Teaching Assessments

Assessments can be used to promote high-quality learning and teaching. Students focus on what they think will be tested. Thus, the selection of proper methods for learning and teaching is crucial.

### 3.3.1. Learning Assessment

**Written assignment.** Written assignments play a crucial role in independent learning, as they offer students the opportunity to reinforce and apply their comprehension of theoretical concepts in written form. By presenting their own views and reflections, students are motivated to think independently and analytically, which helps them gain a comprehensive understanding of the learning material. These assignments, such as reports and essays, are typically connected with the module content. The instructors evaluate the assignments based on assessment criteria, such as critical analysis, content structure, and content. The flexible deadlines for written assignments allow students to work at their own pace (see Figure 2b).

**Project work and demonstration events.** The students project work is evaluated based on weekly deliverables and demonstrations (see Figure 3.2.3 for the example of deliverables). This method of evaluation assesses their teamwork skills and measures their proficiency in ASD. Discussions and interactive events, such as sprint reviews and retrospectives, are used to examine their learning progress. The feedback received during sprint reviews from instructors is valuable, and during the sprint retrospective, students showcase the prototypes and project artifacts they created. Peer assessment is also utilized to encourage other students to ask questions, evaluate the work, and promote discussion.

### 3.3.2. Teaching Assessment

Continuous feedback and final feedback were also used to assess the course. In continuous feedback, students are given the choice of directing their feedback toward the course in general or toward a specific teacher. The final feedback consisted of answers to 14 questions, categorized into different subject areas. The areas were as follows:

1. Learning outcomes and course content
2. Teaching methods and learning assessment
3. Workload, information, and communication

The students rated statements on a Likert scale from strongly agree to strongly disagree. Some questions were choice based and accompanied by supporting questions. Students could clarify their answers in open-text fields. All choice-based questions were mandatory, while open-text questions were optional.

# 4. Empirical Evaluation

The data were collected from the students final feedback gathered over a period of three years: 2021 (n = 14), 2022 (n = 6), and 2023 (n = 43). Their feedback focused on the LOs, course content, teaching methods, and assessment. The data were analyzed descriptively to determine the average values and identify emerging patterns. The qualitative data were further examined to support the quantitative findings. The results of the analysis are depicted in Figure 4. The data indicate that the ratings for all three years were fairly similar, with ratings in between somewhat agree and strongly agree and with slight variations in specific areas, such as course materials, digital tools, and teaching methods. However, it is noteworthy that several aspects showed improvements in 2023.

### 4.0.1. Learning Outcomes and Course Content

The ratings received, which garnered an average of 3.5, indicated that the LOs of the course were

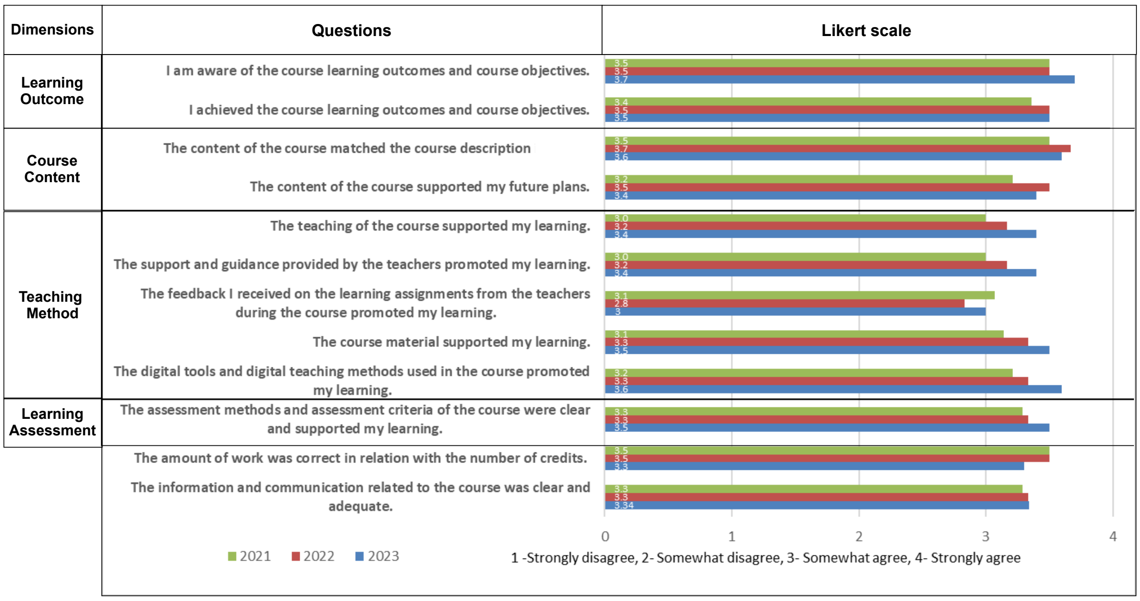


**Figure 4:** Students' feedback on the learning outcomes, course content, teaching methods, and learning assessment (average values)

communicated clearly right from the start. This could be attributed to the clear highlighting of the LOs in Moodle and the effective communication during the course introduction. Additionally, the teaching and assessment methods were designed in such a way that they reflected the achievement of the intended LOs. As one student confirmed,

> Yes, I can say that I learned what I was expecting from this course.

The data also showed that the course material and the expected LOs were well coordinated (rating: 3.03.5), indicating improved content delivery. The course topics were taught in such a way that they integrated the LOs with the learning assessment. The course material was directly relevant to real-world situations or future career goals, which could be attributed to curriculum revisions or good career guidance sessions facilitated by industry professionals. It is recommended that course materials on learning platforms be regularly reviewed and updated to ensure that they are consistent with the latest industry trends and academic research. As one student stated,

> On Moodle, I got every detail related to the assignments and presentation. In fact, guidelines were available from the beginning of the course. This was really helpful for me.

#### 4.0.2. Teaching and Assessment Methods

The teaching methods also received positive feedback, indicating that they improved in 2023. This result could be attributed to the inclusion of class activities, industry lectures, and real-life projects, which contributed to the students positive learning experiences. As two respondents expressed,

> I specifically enjoyed the discussions by the guest lecturers. It was nice to learn about the reality of software development.

The effectiveness of group project work varied, with some students enjoying the collaboration and others facing challenges, such as uneven participation and communication issues. Feedback on the group projects was mixed, with some students appreciating the practical experience and others suggesting improvements in project management and role distribution.

The ratings for digital tools and methods remained around the somewhat agree mark. There is potential to explore more effective digital teaching methods or tools, given the increasing importance of digital tools in education, especially after the pandemic. Some students faced issues with online platforms, the audio quality during lectures, and other technical difficulties, which sometimes hindered the learning process. As one student stated,

> The online lectures were at times absolutely impossible to follow, as the audio quality was atrocious, and even though it did improve over time, it was never good.

The assessments were rated between somewhat agree and strongly agree. The students valued the timely and supportive guidance of their instructors, particularly when it came to comprehending complex topics or managing project work. The students appreciated having clear assessment criteria for their written assignments, but they felt that the project work assessment needed improvement. As one student put it,

> The assessment criteria for the individual tasks were clear and aligned with the learning goals. The group exercise, however, required clear assessment criteria.

## 5. Discussion

### 5.1. Comparing the Learning Outcomes with the Teaching Methods

The course aimed to teach students about professional software engineering processes, specifically ASD. To achieve this goal, the instructors designed LOs (Section 3.1) and used suitable teaching methods (Section 3.2. In-class concepts and practice combined theoretical concepts with practical applications through lectures. Independent work involved completing assignments without direct supervision, while project work linked the participants as they worked actively in groups. These methods helped the students analyze, conceptualize, and evaluate Agile development methods and improvements.

The first research question (Section 1) aimed to explore students evaluations of whether they achieved the intended LOs and their views on the use of research methods in the hybrid setting. Based on the empirical evaluation (Figure 4), the data indicated that the students were satisfied with the teaching methods used. The topics covered during the teaching methods were designed to address the different LOs. The reason for this could be found in Figure 5, which illustrates the teaching methods applied in relation to the LOs. As we can see, the LO objectives were achieved through different teaching method activities. This is why the students confirmed that they achieved the intended LOs in the course.

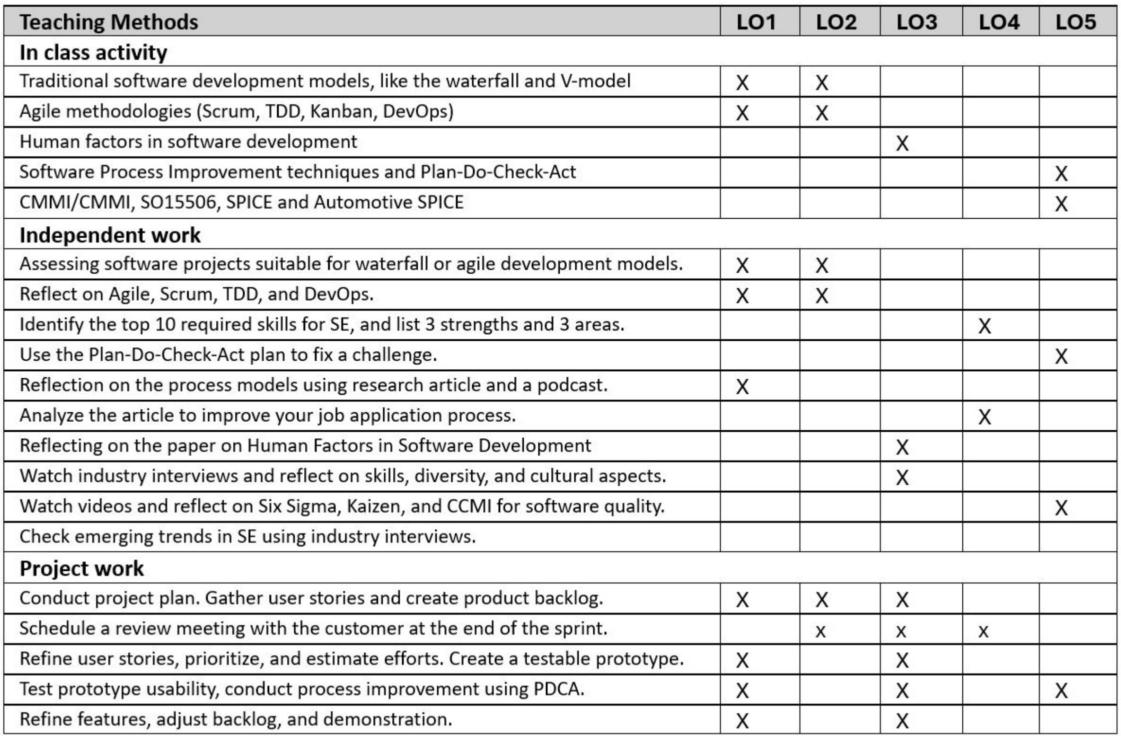
### 5.2. Teaching Methods in Hybrid Settings

As stated in Section 1 through the second question, our goal is to investigate the opportunities associated with hybrid learning. We will draw on our own experiences to provide relevant insights and useful recommendations on effective ways to engage learners in a hybrid learning environment. Our analysis will cover in-class teaching, projects, and independent study. Additionally, we will explore several dimensions in hybrid settings that can significantly affect the efficacy of teaching methods. These dimensions, as mentioned in [16], include space, tools, culture, and coordination.

Space refers to the physical and digital design and utilization of environments that blend remote and in-person activities. Tools refer to the different software, platforms, and technologies that aid collaboration, communication, and productivity in mixed work environments. Culture relates to the shared values, practices, beliefs, and behaviors that shape the social and professional environments of an organization operating in a mix of remote and in-person work arrangements. Coordination involves effectively managing and aligning tasks, projects, and teams operating in a combination of remote and in-person work environments. We discussed each of these to gain more insight into their effects on hybrid learning environments. The findings are outlined in Table 2.

### 5.3. Study Validity

Our course was designed to ensure the studys construct validity by following established pedagogical principles. We utilized constructive alignment and well-known teaching methods and assessments in software engineering education. Survey questions based on constructive alignment principles helped collect the students data aligned with our course LOs, teaching methods, and assessments. To ensure the studys external validity, the content referred to previous literature on ASD courses and included essential topics. The course was adapted for over three years to suit hybrid learning situations, making it useful for other educators and learners in similar settings. To address reliability, we published our survey questions for other researchers to utilize in their own studies and to help them achieve comparable outcomes. The authors worked together to develop the course, and they integrated their knowledge of hybrid teaching.



| Teaching Methods | LO1 | LO2 | LO3 | LO4 | LO5 |
|---|---|---|---|---|---|
| **In class activity** | | | | | |
| Traditional software development models, like the waterfall and V-model | X | X | | | |
| Agile methodologies (Scrum, TDD, Kanban, DevOps) | X | X | | | |
| Human factors in software development | | | X | | |
| Software Process Improvement techniques and Plan-Do-Check-Act | | | | | X |
| CMMI/CMMI, SO15506, SPICE and Automotive SPICE | | | | | X |
| **Independent work** | | | | | |
| Assessing software projects suitable for waterfall or agile development models. | X | X | | | |
| Reflect on Agile, Scrum, TDD, and DevOps. | X | X | | | |
| Identify the top 10 required skills for SE, and list 3 strengths and 3 areas. | | | | X | |
| Use the Plan-Do-Check-Act plan to fix a challenge. | | | | | X |
| Reflection on the process models using research article and a podcast. | X | | | | |
| Analyze the article to improve your job application process. | | | | X | |
| Reflecting on the paper on Human Factors in Software Development | | | X | | |
| Watch industry interviews and reflect on skills, diversity, and cultural aspects. | | | X | | |
| Watch videos and reflect on Six Sigma, Kaizen, and CCMI for software quality. | | | | | X |
| Check emerging trends in SE using industry interviews. | | | | | |
| **Project work** | | | | | |
| Conduct project plan. Gather user stories and create product backlog. | X | X | X | | |
| Schedule a review meeting with the customer at the end of the sprint. | | x | x | x | |
| Refine user stories, prioritize, and estimate efforts. Create a testable prototype. | X | | X | | |
| Test prototype usability, conduct process improvement using PDCA. | X | | X | | X |
| Refine features, adjust backlog, and demonstration. | X | | X | | |

**Figure 5:** Mapping between the teaching methods and the learning outcomes.

**Table 2**
Issues and recommendations on teaching methods in hybrid settings

| Aspects | Space | Tools | Culture | Coordination |
|---|---|---|---|---|
| **In-class concepts** | Using omnidirectional microphones and cameras to capture the entire classroom and maintaining eye contact with both groups of students during lectures can balance attention between in-person and remote learners. | Online learning platforms can face technical issues that disrupt hybrid learning. Conducting small training sessions on common issues during course introduction can help address technical problems. | Using interactive tools, such as polls, quizzes, and breakout rooms, can ensure equal engagement for in-person and remote students to enable a common understanding of course topics. | Scheduling lecture times with remote students across different time zones can be challenging. Providing learning resources before class sessions and sharing the video recordings of lectures can provide remote groups with equal access. |
| **Project work** | The use of digital collaboration tools, such as Mural, Miro, and Trello, to create a shared workspace can help ease the challenges of collaborating with both in-person and remote students. | Uneven tool knowledge may hinder progress in groups. To familiarize members with the tools being used, instructors can provide recorded training videos and resources at the beginning of the course. | To promote trust and a positive team culture, instructors should encourage hybrid teams to adopt good practices and conduct regular check-ins. | Remote team members often face challenges when scheduling across time zones. Tools, such as Doodle and When2meet, can be used to find common meeting times. |
| **Independent study** | Remote students may face difficulty in completing tasks. They should be encouraged to create a dedicated workspace at home. | Students' progress can be hindered by technical issues. Instructors can provide deadline flexibility to address these challenges. | Remote students require support to stay motivated and engaged. Encouraging peer interaction through discussion forums and regular check-ins can help students remain engaged in their studies. | Remote students may need help from their instructors and require guidance on effective time management strategies. Structuring the course with regular milestones can help students stay on track. |

## 6. Conclusion

This study examined how a course on ASD and software processes can be conducted, especially when students learn in person and remotely. The research attempted to understand the unique features of and the possible problems with this type of teaching. It asked important questions about what the students thought of the hybrid courses and how teachers could overcome the challenges they faced. The knowledge obtained from this study is relevant to Agile researchers and teachers because it uses real experiences from teaching courses and a careful analysis of student feedback collected over three years. This information can help teachers improve their classes and give students more valuable learning experiences. The study also identified the problems that teachers might face in hybrid classes and suggested ways to address them. Overall, the findings can provide practical knowledge for teachers looking to improve their classes and ensure that all students, regardless of the learning mode, can learn effectively.

## References

[1] E. Ceh-Varela, C. Canto-Bonilla, D. Duni, Application of project-based learning to a software engineering course in a hybrid class environment, Information and Software Technology 158 (2023) 107189.

[2] R. Verdecchia, P. Lago, Tales of hybrid teaching in software engineering: Lessons learned and guidelines, IEEE Transactions on Education (2022).

[3] M. Moster, D. Ford, P. Rodeghero, " is my mic on?" preparing se students for collaborative remote work and hybrid team communication, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), IEEE, 2021, pp. 89–94.

[4] I. Noguera, A.-E. Guerrero-Roldán, R. Masó, Collaborative agile learning in online environments: Strategies for improving team regulation and project management, Computers & Education 116 (2018) 110–129.

[5] M. Neumann, L. Baumann, Agile methods in higher education: Adapting and using eduscrum with real world projects, in: 2021 IEEE Frontiers in Education Conference (FIE), IEEE, 2021, pp. 1–8.

[6] J. H. Sharp, G. Lang, Agile in teaching and learning: Conceptual framework and research agenda, Journal of Information Systems Education 29 (2018) 45–52.

[7] T. F. Otero, R. Barwaldt, L. O. Topin, S. V. Menezes, M. J. R. Torres, A. L. de Castro Freitas, Agile methodologies at an educational context: a systematic review, in: 2020 IEEE Frontiers in Education Conference (FIE), IEEE, 2020, pp. 1–5.

[8] Z. Masood, R. Hoda, K. Blincoe, Adapting agile practices in university contexts, Journal of Systems and Software 144 (2018) 501–510.

[9] J. Biggs, Enhancing teaching through constructive alignment, Higher education 32 (1996) 347–364.

[10] O. Hyppönen, S. Lindén, et al., Handbook for teachers: course structures, teaching methods and assessment (2009).

[11] T. Hynninen, J. Kasurinen, A. Knutas, O. Taipale, Guidelines for software testing education objectives from industry practices with a constructive alignment approach, in: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, 2018, pp. 278–283.

[12] A. Cain, M. A. Babar, Reflections on applying constructive alignment with formative feedback for teaching introductory programming and software architecture, in: Proceedings of the 38th International Conference on Software Engineering Companion, 2016, pp. 336–345.

[13] N. B. Ruparelia, Software development lifecycle models, ACM SIGSOFT Software Engineering Notes 35 (2010) 8–13.

[14] M. Behroozi, S. Shirolkar, T. Barik, C. Parnin, Debugging hiring: What went right and what went wrong in the technical interview process, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society, 2020, pp. 71–80.

[15] E. Dutra, B. Diirr, G. Santos, Human factors and their influence on software development teams-a tertiary study, in: Proceedings of the XXXV Brazilian Symposium on Software Engineering, 2021, pp. 442–451.

[16] M. Grzegorczyk, M. Mariniello, L. Nurski, T. Schraepen, Blending the physical and virtual: a hybrid model for the future of work, Technical Report, Bruegel Policy Contribution, 2021.